

| Claim   | BlackBerry Internet Content Functional Specification Document (Rev. 1.3)  |
|---|---|
| <p>1. A system for retrieving information from one or more information sources, wherein the information includes a plurality of content types, comprising:</p> <p>a host device coupled to the one or more information sources for retrieving the information; and</p> <p>a translation component coupled to the host device, the translation component including a plurality of content translators, wherein each content translator is configured to translate one of the plurality of content types into common virtual machine language programs.</p> | <p>Section 7: Host Development (p. 11) shows a system for retrieving information from one or more information sources ("Target Web Sites"), wherein the information may include a plurality of content types. Section 4: Complexity Questions (p. 6) describes various content types that may be supported by the system, including HTML, XML, WML, WML Scripting, Visual Basic Scripting, HDML, Java, etc. Also, Section 3: User Experience (p. 4) states that "The types of languages RIM chooses to support is completely up to what has been built into the proxy server for translation."</p> <p>Section 7: Host Development (p. 11) shows a Linux Proxy Machine, which is a host device coupled to a the Target Web Sites (information sources) for retrieving information.</p> <p>Section 7: Host Development (pp. 11) shows a block labeled "Content Filter &amp; Transformation," or "Filters &amp; Transformation Engine." This component performs a transformation or translation of the information into byte code programs. From Section 7.1 (p. 12) "The filter and transformation engine... performs a byte-code transformation on the information so that it can be executed as a program on the device." Section 4: Complexity Questions pp. 5-6), "Another way to achieve this functionality is to send down programs or applets, like a Java Applet, that would support a programming language similar to Java that the developer creates as part of a special URL access points." Also in the same section, "By using this byte code method it would be possible to convert the 'browser' on the device to a program interpreter. The program interpreter would execute the downloaded instructions and do whatever the end-developer instructed." The document clearly described a plurality of these Filter &amp; Transformation Engines for each content type handled by the Proxy Machine. From Section 7.2 (p. 12) "On the host side [proxy] of the connection, probably in close proximity to the SWS interface code, will be a series of filters and byte-code generators." Also from the same section, "The goal of the filters will be to take the raw HTML, WML, WML Scripting, Java or other content from the Internet and strip out any content that cannot be turned into byte-code instructions. In some cases more than one filter may have to be executed if the content includes different components, i.e., WML with WML Scripting embedded." The transformation function discussed in Section 7 is previously mentioned in Sections 3 and 4 as</p> |

RECEIVED  
DEC 01 2004  
Technology Center 2100

| Claim   | BlackBerry Internet Content Functional Specification Document (Rev. 1.3)  |
|---|---|
|   | <p>being a translation function. These two terms are used interchangeably in the document.</p> <p>Section 3: User Experience (p. 4) states that “The types of languages RIM chooses to support is completely up to what has been built into the proxy server for translation.” Section 4: Complexity Questions (p. 6) “These advanced formats and languages could all be translated into the RIM proprietary byte code.” The byte code generators create the common virtual machine language program that are executed by the program interpreter on the device. (Section 6: Internet Browser)</p>  |
| <p>2. The system of claim 1, further comprising:</p> <p>a plurality of client devices coupled to the translation component, each of the client devices including an interface for receiving common virtual machine language programs from the translation component and a virtual machine engine for executing the received common virtual machine language programs.</p> | <p>Section 7: Host Development (p. 11) shows an example client side device, which is clearly one of a plurality of client side devices that can be used with the system, and includes an interface (Radio Code/OS, Transport Layer, Secure Transport – GME Layer) for receiving virtual machine language programs (which are also called byte code programs in the document). Also included on the client side device is the virtual machine language engine (referred to as the Program Interpreter), which is the code that executes the downloaded byte code programs.</p> <p>Section 4: Complexity Questions pp. 5-6), “Another way to achieve this functionality is to send down programs or applets, like a Java Applet, that would support a programming language similar to Java that the developer creates as part of a special URL access points.” Java programs are executed by a Java virtual machine. Also in the same section, “By using this byte code method it would be possible to convert the ‘browser’ on the device to a program interpreter. The program interpreter would execute the downloaded instructions and do whatever the end-developer instructed.” Section 9: Conclusions (p. 16), “This program interpreter will provide the ‘effect’ of the browser environment by turning HTML content into program interpretive commands.”</p> <p>Section 8: Content Provider (p. 15) “One of the most powerful aspects of running a byte-code interpreter on the Blackberry handheld is that we can delivery programs to the pager for execution. The initial offering will allow ‘partial’ Java programs to be sent to the pager for execution.”</p> |

| Claim  | BlackBerry Internet Content Functional Specification Document (Rev. 1.3)   |
|--|--|
| <p>3. The system of claim 2, wherein the plurality of client devices further include a file explorer component for sending an information request to the host device, which then retrieves the requested information from the one or more information sources and passes the requested information to the translation component.</p> | <p><b>Section 6: Internet Browser (p. 8), “There are two possible designs for the Internet Browser, which offer very different functionality. . . The second design is not a browser at all but a file explorer and program interpreter. RIM has decided to develop the program interpreter that can behave like a browser.” Section 6.2: Program Interpreter (P. 8), “In this second design the new program is a Program Interpreter with two main roles. The first role is to present the available programs to be executed and allow them to maneuver through the files available; similar to viewing the cache. The second role is to load and execute the programs as requested by the user, or as fetched from a web site.” Section 9: Conclusions (p. 16), “To manage a collection of loaded programs, or web pages stored as programs, the Command Interpreter has an ‘explorer like’ interface for launching and fetching these items. . . The result is a HTML/Java experience on the pager and for the developer of content sent to the pager.”</b></p> |
| <p>4. The system of claim 1, wherein at least one of the plurality of content translators is an HTML content translator for translating HTML type content into a virtual machine language program.</p>   | <p><b>Section 4: Complexity Questions (p. 6) describes various content types that may be supported by the system, including HTML, XML, WML, WML Scripting, Visual Basic Scripting, HDML, Java, etc.</b></p>  |
| <p>5. The system of claim 1, wherein at least one of the plurality of content translators is an HDML content translator for translating HDML type content into a virtual machine language program.</p>   | <p><b>Section 4: Complexity Questions (p. 6) describes various content types that may be supported by the system, including HTML, XML, WML, WML Scripting, Visual Basic Scripting, HDML, Java, etc.</b></p>  |

| Claim  | BlackBerry Internet Content Functional Specification Document (Rev. 1.3)  |
|--|---|
| <p>6. The system of claim 1, wherein at least one of the plurality of content translators is an XML content translator for translating XML type content into a virtual machine language program.</p>   | <p><b>Section 4: Complexity Questions (p. 6) describes various content types that may be supported by the system, including HTML, XML, WML, WML Scripting, Visual Basic Scripting, HDML, Java, etc.</b></p> |
| <p>7. The system of claim 1, wherein at least one of the plurality of content translators is a WML content translator for translating WML type content into a virtual machine language program.</p>  | <p><b>Section 4: Complexity Questions (p. 6) describes various content types that may be supported by the system, including HTML, XML, WML, WML Scripting, Visual Basic Scripting, HDML, Java, etc.</b></p> |
| <p>8. The system of claim 1, wherein the plurality of content translators include an HTML content translator and an XML content translator, wherein each of the HTML and XML content translators translates content information into a common virtual machine language program.</p>  | <p><b>Section 4: Complexity Questions (p. 6) describes various content types that may be supported by the system, including HTML, XML, WML, WML Scripting, Visual Basic Scripting, HDML, Java, etc.</b></p> |
| <p>9. The system of claim 1, wherein the plurality of content translators include an HTML content translator, an XML content translator, and an HDML content translator, wherein each of the HTML, XML and HDML content translators translates content information into a common</p> | <p><b>Section 4: Complexity Questions (p. 6) describes various content types that may be supported by the system, including HTML, XML, WML, WML Scripting, Visual Basic Scripting, HDML, Java, etc.</b></p> |

| Claim   | BlackBerry Internet Content Functional Specification Document (Rev. 1.3)  |
|---|---|
| virtual machine language program.   |   |
| 10. The system of claim 1, wherein the information is retrieved from the one or more information sources through a TCP/IP network connection between the host system and the one or more information sources. | <b>Section 7: Host Development (p. 11) shows that the host system (Linux Proxy Machine) is connected to multiple Target Web Sites using an HTTP Fetch &amp; Cache Component. HTTP is the higher level protocol used to access Web Sites over the Transmission Control Protocol/Internet Protocol (TCP/IP) network connection.</b> |
| 11. The system of claim 10, wherein at least one of the one or more information sources is a web site accessible via the Internet.  | <b>Section 7: Host Development (p. 11) shows that the host system (Linux Proxy Machine) is connected to multiple Target Web Sites using an HTTP Fetch &amp; Cache Component. HTTP is the higher level protocol used to access Web Sites over the Transmission Control Protocol/Internet Protocol (TCP/IP) network connection.</b> |
| 12. The system of claim 1, further comprising:<br>a byte code generator coupled to the translation component for compressing the common virtual machine language programs.                                    | <b>Section 7.2: Host Filtering Requirements (p. 12) shows the byte-code generator coupled to one of the filter and transformation blocks that convert the downloaded content into virtual machine language programs.</b>  |
| 13. (Cancelled)   |   |
| 14. The system of claim 2, further comprising a network coupling the plurality of client devices to the translation component.  | <b>Section 7: Host Development (p. 11) shows a wireless network coupling the clients to the Linux Proxy Machine – the host system where the translation elements are located.</b>   |
| 15. The system of claim 14, wherein the translation component   | <b>Section 7: Host Development (p. 11) shows several network protocol interface elements (SWS Agent and SRP Link, and SWS Machine – GME Layer) that enable the translation components</b>   |

| <b>Claim</b>   | <b>BlackBerry Internet Content Functional Specification Document (Rev. 1.3)</b>   |
|--|---|
| further includes a network protocol interface for transmitting and receiving data via the network.   | (Filters & Transformation Engine) to transmit and receive data via the wireless network.  |
| 16. The system of claim 15, wherein the network protocol interface packetizes the virtual machine language programs for transmission to the plurality of client devices via the network. | The elements SWS Agent and SRP Link and GME Layer shown in Section 7 packetize the virtual machine language programs for transmission over the wireless network.                      |
| 17. The system of claim 14, wherein the network is a wireless digital data network.  | Section 7: Host Development (p. 11) shows a wireless network.   |
| 18. The system of claim 17, wherein the wireless digital data network is a packet data network.  | Section 7: Host Development (p. 11) shows a wireless network.   |
| 19. The system of claim 2, wherein the plurality of client devices are mobile computing apparatuses.   | The client devices shown in the document are clearly mobile computing apparatuses.  |
| 20. The system of claim 19, wherein the mobile computing apparatuses are PDAs, cellular telephones, Internet appliances, or two-way pagers.  | The Blackberry Handheld client mentioned in the document as a client device is a two-way pager. Section 9: Conclusions (p. 16) "The result is a HTML/Java experience on the pager..." |
| 21. (Cancelled)  |   |
|  |   |

| Claim  | BlackBerry Internet Content Functional Specification Document (Rev. 1.3)   |
|--|--|
| 22. The system of claim 1, wherein the content translator is located at the host system.   | Section 7: Host Development (P. 11) shows the content translator (Content Filter & Transformation) located at the Linux Proxy Machine.   |
| 23. The system of claim 1, wherein the content translator is coupled to the host system via an HTTP interface.   | Section 7: Host Development (p. 11), "In this example the Content Filter could live within the Linux environment." The Figure labeled "Host Side Components" shows an HTTP interface coupled to the Filter and Transformation Engine.  |
| 24. The system of claim 2, wherein the plurality of client devices further include a program store for storing a plurality of virtual machine language programs received from the translation component.   | The Blackberry client devices include a cache memory for storing the downloaded byte code programs. Section 6.2: Program Interpreter (p. 8) "The second role is to load and execute the programs as requested by the user, or as fetched from a web site." (p. 9) "The user can also get access to saved Gizmos [See Section 2: Terminology] similar to saved messages." The saved Gizmos are the downloaded byte code programs. Section 3: User Experience (p. 4) "They can browse through a cache of locally saved files, like Windows Explorer, or eve select a completely random file (URL) to download or fetch."   |
| 25. The system of claim 24, wherein the plurality of client devices further include a file explorer and storage interface for generating an information request, determining whether the requested information is associated with a corresponding virtual machine language program stored in the program store, and if so, then retrieving the virtual machine language program from the program store, and if not, then sending the information request to the host system. | <p>Section 6: Internet Browser (p. 8), "There are two possible designs for the Internet Browser, which offer very different functionality. . . The second design is not a browser at all but a file explorer and program interpreter. RIM has decided to develop the program interpreter that can behave like a browser." Section 6.2: Program Interpreter (P. 8), "In this second design the new program is a Program Interpreter with two main roles. The first role is to present the available programs to be executed and allow them to maneuver through the files available; similar to viewing the cache. The second role is to load and execute the programs as requested by the user, or as fetched from a web site." Section 9: Conclusions (p. 16), "To manage a collection of loaded programs, or web pages stored as programs, the Command Interpreter has an 'explorer like' interface for launching and fetching these items. . . The result is a HTML/Java experience on the pager and for the developer of content sent to the pager."</p> <p>The Blackberry client devices include a cache memory for storing the downloaded byte code</p> |

| Claim           | BlackBerry Internet Content Functional Specification Document (Rev. 1.3)  |
|-----------------|---|
|                 | <p>programs. Section 6.2: Program Interpreter (p. 8) “The second role is to load and execute the programs as requested by the user, or as fetched from a web site.” (p. 9) “The user can also get access to saved Gizmos [See Section 2: Terminology] similar to saved messages.” The saved Gizmos are the downloaded byte code programs. Section 3: User Experience (p. 4) “They can browse through a cache of locally saved files, like Windows Explorer, or even select a completely random file (URL) to download or fetch.”</p> <p>Section 6.2: Program Interpreter (p. 10) “When the user selects the item, by double clicking or by hitting enter, a ‘loading’ screen will be presented, shown in the illustration. The loading screen may only be present for seconds, unless the program must be fetched first or if the program is very large.”</p> |
| 26. (Cancelled) |   |